Gui Frameworks

Table of Contents

1. Introduction . . . 1
2. Qt . . . 1
2.1. Create simple application without Qt Creator . . . 1
2.2. Deploy Qt application . . . 3
2.2.1. Windows . . . 3
2.3. Troubleshooting . . . 4

1. Introduction

Gui Frameworks known also as widget toolkits are used to construct Graphical User Interface programs.

2. Qt

Doc Writer <christian.popescu@outlook.com> v 1.1, 2020-12-08 :toc:

Qt is a cross-platform application development framework for desktop, embedded and mobile.

With **Qt**, GUIs can be written directly in C++ using its **Widgets** module. Qt also comes with an interactive graphical tool called **Qt Designer** which functions as a code generator for Widgets based GUIs. **Qt Designer** can be used stand-alone but is also integrated into Qt Creator.

Qt is far more than a GUI toolkit. It provides modules for cross-platform development in the areas of networking, databases, OpenGL, web technologies, sensors, communications protocols (Bluetooth, serial ports, NFC), XML and JSON processing, printing, PDF generation, and much more.

2.1. Create simple application without Qt Creator

1. Create project folder

```
Example: 03-SimpleAppWithouQtCreator
```

2. Add the following main.cpp file

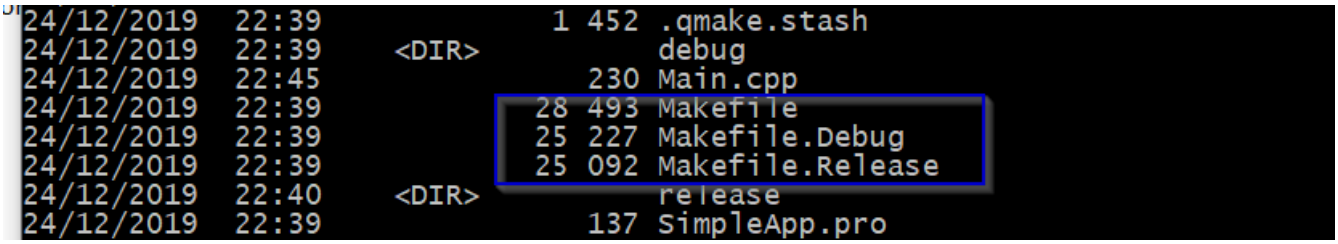
```
#include <QApplication>
#include <QColor>
#include <QTextEdit>

int main(int argc, char **argv)
{
 QApplication app (argc, argv);

 QTextEdit textEdit("My Text in text edit");
 textEdit.show();

 return app.exec();
}
```

3.Add Qt project file SimpleApp.pro

```
TEMPLATE = app
TARGET = simpleappwithoutQt

QT = core gui

greaterThan(QT_MAJOR_VERSION, 4): QT += widgets

SOURCES +=  main.cpp
```

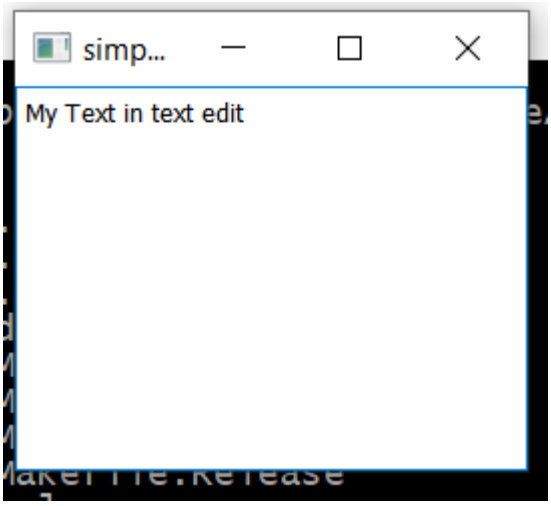
4.In QT command window (MSVS or MingW) call **qmake** tool. The qmake generates the make files.

Note: In MSVS the vcvars.bat should be run before.



5.In QT command windows run the make/nmake command.

6.Run the *.exe file created in release/debug folder (from command line).



2.2. Deploy Qt application

2.2.1. Windows

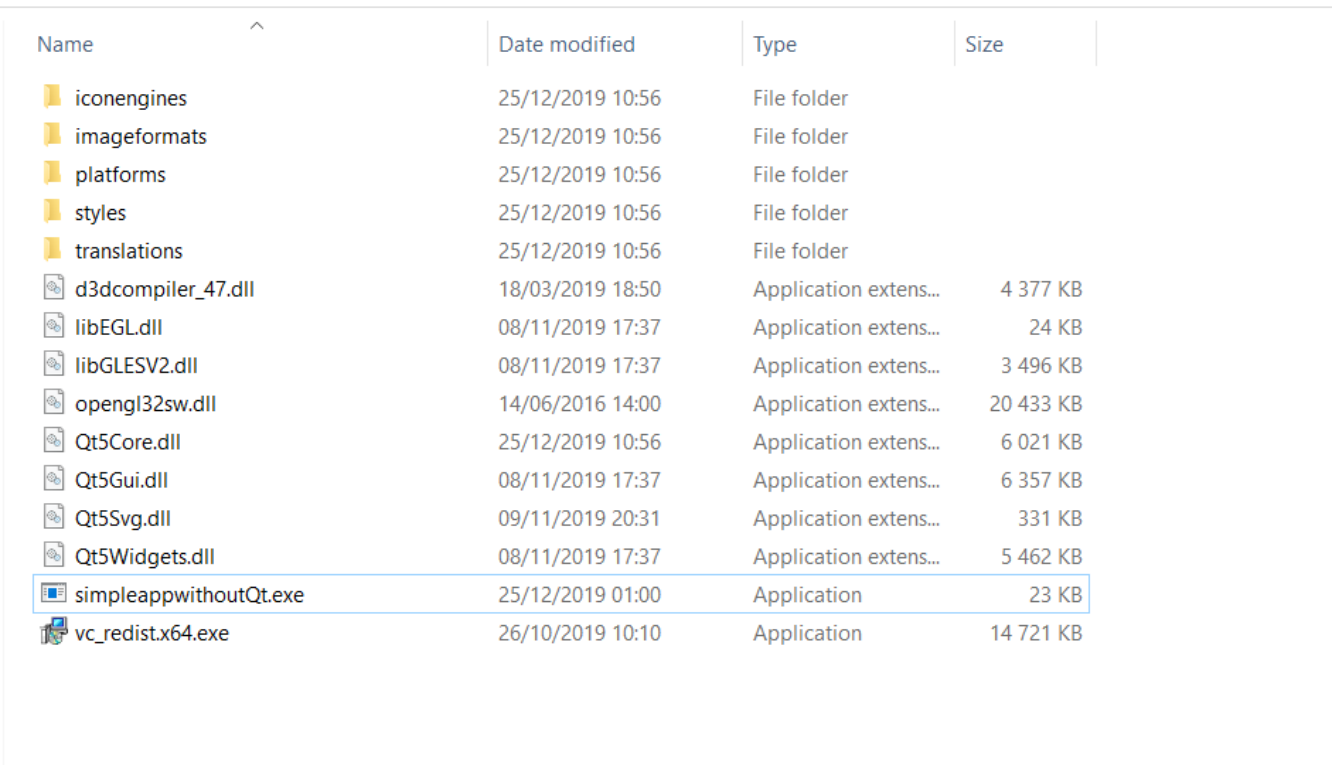
Qt application Deployement for Windows

1. Execute nmake release. This will create the **release** folder.
2. In the release folder execute **windeployqt** windows deployement tool.

```
Example: windeployqt simpleappwithoutQt.exe
```

```
E:\ccp_vhdd_main\workspace_ming\Poc\03-QtSimpleApp\04-SimpleAppSeparateWindow\release>windeployqt simpleappwithoutQt.exe
E:\ccp_vhdd_main\workspace_ming\Poc\03-QtSimpleApp\04-SimpleAppSeparateWindow\release\simpleappwithoutQt.exe 64 bit, release executable
Adding Qt5Svg for qsvgicon.dll
Direct dependencies: Qt5Core Qt5Widgets
All dependencies   : Qt5Core Qt5Gui Qt5Widgets
To be deployed     : Qt5Core Qt5Gui Qt5Svg Qt5Widgets
Updating Qt5Core.dll.
Updating Qt5Gui.dll.
Updating Qt5Svg.dll.
Updating Qt5Widgets.dll.
Updating libGLESV2.dll.
Updating libEGL.dll.
Updating d3dcompiler_47.dll.
Updating opengl32sw.dll.
Updating vc_redist.x64.exe.
Patching Qt5Core.dll...
Creating directory E:/ccp_vhdd_main/workspace_ming/Poc/03-QtSimpleApp/04-SimpleAppSeparateWindow/release/iconengines.
Updating qsvgicon.dll.
Creating directory E:/ccp_vhdd_main/workspace_ming/Poc/03-QtSimpleApp/04-SimpleAppSeparateWindow/release/imageformats.
Updating qgif.dll.
Updating qicns.dll.
Updating qico.dll.
Updating qjpeg.dll.
Updating qsvg.dll.
Updating qtga.dll.
Updating qtiff.dll.
Updating qwbmp.dll.
Updating qwebp.dll.
Creating directory E:/ccp_vhdd_main/workspace_ming/Poc/03-QtSimpleApp/04-SimpleAppSeparateWindow/release/platforms.
Updating qwindows.dll.
Creating directory E:/ccp_vhdd_main/workspace_ming/Poc/03-QtSimpleApp/04-SimpleAppSeparateWindow/release/styles.
Updating qwindowsvistastyle.dll.
Creating E:\ccp_vhdd_main\workspace_ming\Poc\03-QtSimpleApp\04-SimpleAppSeparateWindow\release\translations...
Creating qt_ar.qm...
Creating qt_bg.qm...
Creating qt_ca.qm...
Creating qt_cs.qm...
Creating qt_da.qm...
Creating qt_de.qm...
Creating qt_en.qm...
Creating qt_es.qm...
Creating qt_fi.qm...
Creating qt_fr.qm...
Creating qt_gd.qm...
Creating qt_he.qm...
Creating qt_hu.qm...
Creating qt_it.qm...
Creating qt_ja.qm...
Creating qt_ko.qm...
Creating qt_lv.qm...
Creating qt_pl.qm...
Creating qt_ru.qm...
Creating qt_sk.qm...
Creating qt_uk.qm...
Creating qt_zh_TW.qm...
```

3.This creates the deployement package.



Name	Date modified	Type	Size
iconengines	25/12/2019 10:56	File folder	
imageformats	25/12/2019 10:56	File folder	
platforms	25/12/2019 10:56	File folder	
styles	25/12/2019 10:56	File folder	
translations	25/12/2019 10:56	File folder	
d3dcompiler_47.dll	18/03/2019 18:50	Application extens...	4 377 KB
libEGL.dll	08/11/2019 17:37	Application extens...	24 KB
libGLESV2.dll	08/11/2019 17:37	Application extens...	3 496 KB
opengl32sw.dll	14/06/2016 14:00	Application extens...	20 433 KB
Qt5Core.dll	25/12/2019 10:56	Application extens...	6 021 KB
Qt5Gui.dll	08/11/2019 17:37	Application extens...	6 357 KB
Qt5Svg.dll	09/11/2019 20:31	Application extens...	331 KB
Qt5Widgets.dll	08/11/2019 17:37	Application extens...	5 462 KB
simpleappwithoutQt.exe	25/12/2019 01:00	Application	23 KB
vc_redist.x64.exe	26/10/2019 10:10	Application	14 721 KB

2.3. Troubleshooting

Sometimes the build 64bits doesn't work.

In this case check QMAKE_SPEC variable and set it as win64-msvc

- to query

```
qmake -query QMAKE_SPEC
```

- to set

```
qmake -set QMAKE_SPEC win64-msvc
```